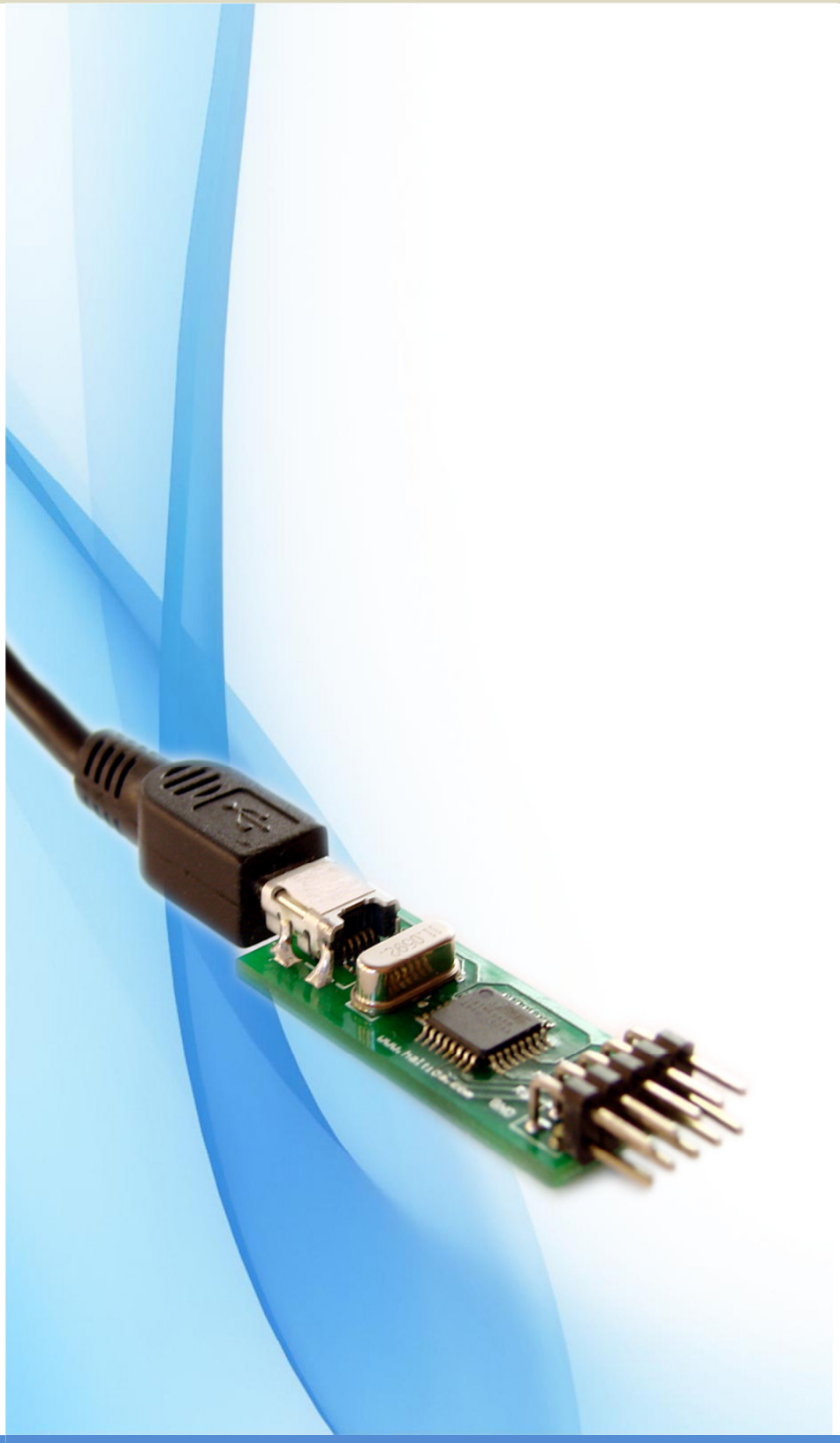
HALTICA



Phantom

Manual de Usuario

soporte@haltica.com

Phantom
Manual de Usuario

Versión 1.1 - Mayo 2012

Versión 1.0 - Mayo 2009

Contenido.

Capítulo 1	**Phantom.**	**1**
	Descripción Requerimientos mínimos Dispositivos soportados	
Capítulo 2	**Conexión del equipo**	**3**
	Conexiones de Hardware	
Capítulo 3	**Instalación de Drivers**	4
	Descarga de drivers - Drivers comprimidos en formato .zip - Drivers con autoejecutable	
Capítulo 4	**Uso con Atmel Studio 6**	5
	Verificaci;on del puerto COM asignado Conexión con ATMEL Studio 6 Lectura de firma de dispositivo Programación de la memoria Flash/EEPROM del microocntrolador Fusibles y Lock bits	

[página en blanco]

1 PHANTOM

Gracias por adquirir el programador Phantom de Haltica.

Antes de comenzar por favor lea este manual para evitar un daño a su persona o a el equipo que este utilizando.

1.1 Descripción.

Phantom es un programador de conexión USB a la computadora, programación ISP para microcontroladores AVR de ATMEL y que cuenta con interface directa entre AVR Studio 4 y Atmel Studio 6[I].

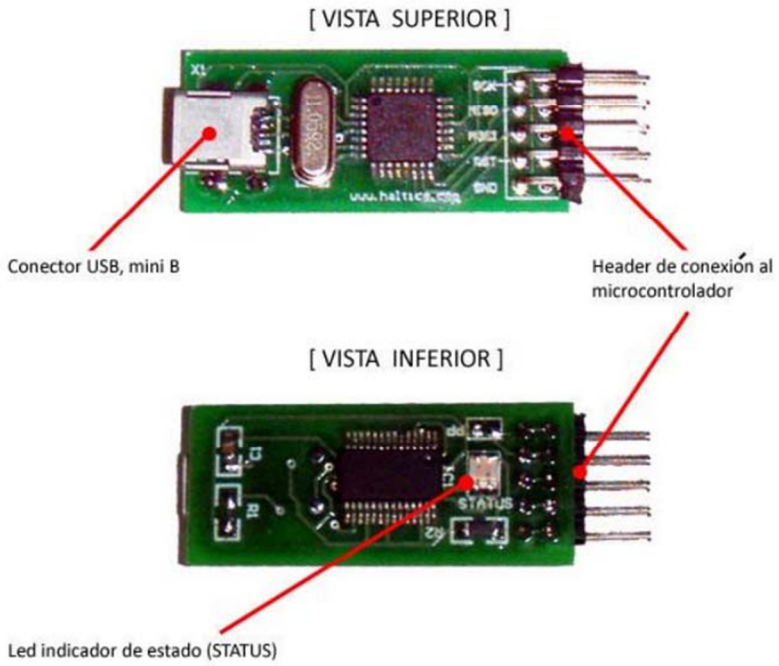


Figura 1. Detalles del Phantom

- Conector USB: Utilizado para conectarse a la computadora con el cable USB A a mini B.
- Header de conexión: facilita su colocación directamente en un protoboard o bien a través de un cable plano (no proporcionado con el equipo). Es aquí donde se tiene acceso a las terminales de programación: MOSI, MISO, SCK y RESET así como al pin de GND necesario para que tanto el voltaje del programador (proporcionado por

la computadora, vía USB) como el voltaje del microcontrolador a programar (requiere de una fuente externa), estén al mismo potencial.

- LED de estatus: Indica el estado del programador, este puede variar dependiendo de la versión de firmware del equipo. Puede ser bicolor o blanco.

Tenga en cuenta que las terminales de programación ISP (header 2x5) no cuentan con aislamiento eléctrico o dispositivos especializados para protección ante descargas electrostáticas. Por tal motivo es responsabilidad del usuario verificar las conexiones que el mismo realice, así como los niveles de tensión y corriente que le aplique al dispositivo para evitar dañar el equipo o su propia persona.

1.2. Requerimientos mínimos.

Usted requiere los siguientes elementos para poder utilizar el programador Phantom:

1 Programador Phantom,
1 cable USB A a mini B para la conexión con la computadora,
1 computadora con sistema operativo[II] (Windows XP/Vista/7 de 32 o 64 bits) que cuente con el software Atmel Studio 6 o AVR Studio 4, los drivers del Phantom instalados y una bahía USB disponible.
1 Circuito con algún AVR soportado por el programador, correctamente conectado y con su propia fuente de alimentación.

1.3. Dispositivos soportados.

Phantom es capaz de programar la gran mayoría de los microcontroladores AVR de 8 bits (las únicas excepciones conocidas son: AT90S1200 y ATMega103, ambos ya descontinuados) que cuenten con programación ISP de las familias: AT90S, Tiny, Mega, USB, CAN y PWM.

[I] También funciona con las versiones de AVR Studio 5 y 5.1, el proceso es el mismo que el descrito en este documento para ATMEL Studio 6
[II] También puede utilizar una máquina virtual que cuente con alguno de esos sistemas operativos instalados, sin embargo no se ha probado exhaustivamente el funcionamiento del equipo bajo esas condiciones.

2 CONEXIÓN DEL EQUIPO

2.1. Conexiones de Hardware.

La conexión del phantom hacia la computadora se realiza a través del puerto USB mini B, utilizando el cable USB A a mini B (suministrado con el equipo) o similar.

Las conexiones entre el phantom y el AVR a programar se realizarán en paralelo, una a una como se muestra en la siguiente figura. Para facilitar la identificación de cada una de las terminales, el nombre de cada una se encuentra impreso en la serigrafía del phantom como se muestra en la siguiente figura.

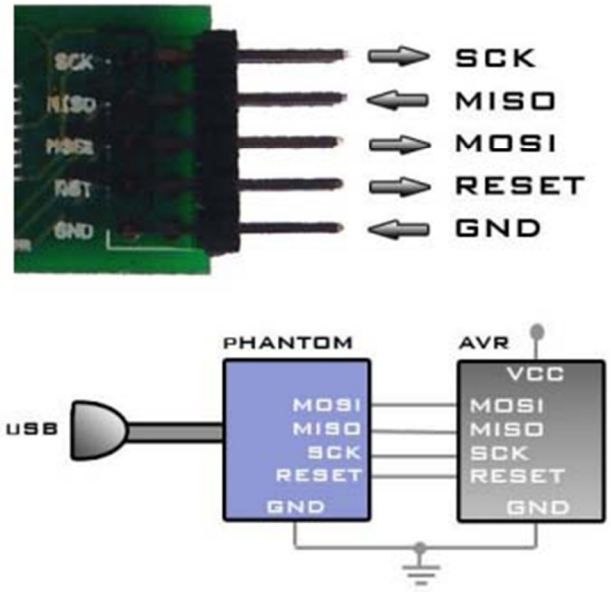


Figura 2.1 Terminales de programación y conexión con un AVR.

Tanto el programador Phantom como los microcontroladores AVR cotienen o son dispositivos CMOS, por tal motivo debe prestar atención y cuidado en el manejo de los mismos para evitar algun daño por descargas electrostáticas.

3 INSTALACIÓN DE DRIVERS

La primera vez que conecte el phantom a la computadora, le pedirá que instale los drivers correspondientes (excepto si previamente ya los había instalado).

3.1. Descarga de drivers.

Los drivers los puede descargar de la página de haltica: http://www.haltica.com en la sección de productos, phantom.

Existen dos versiones, que funcionan de la misma manera la diferencia es el modo de instalación, elija el que mas le convenga.

3.1.1. Drivers comprimidos en formato .zip.

Algunos equipos con procesador Intel Atom han presentado dificultades en el uso de los drivers con autoejecutable, para resolver ese problema puede utilizar esta opción.

Descarge y descomprima en una ruta conocida (por ejemplo: cree una carpeta en el escritorio llamada drivers). Conecte el programador, espere por el mensaje de Windows pidiendo los drivers, seleccione la opción de instalar desde una ubicación específica e indique la ubicación de la carpeta donde descomprimió los drivers (en nuestro ejemplo esta se encontraría en la carpeta drivers ubicada en el escritorio).

Es posible que le pida realizar este procedimiento dos veces.

3.1.2. Drivers con autoejecutable.

Esta es la opción mas sencilla, simplemente descargue y ejecute el programa. En esta opción no es necesario que el programador este conectado para poder ejecutar el instalador.

4 USO CON ATMEL STUDIO 6

Para utilizar el Phantom directamente con el ATMEL Studio 6 usted necesita tener instalados los drivers (para más información vea el capitulo 3: Instalación de Drivers).

Para la correcta realización de los siguientes pasos usted requiere un circuito que cuente con un AVR, (en el caso de este ejemplo, el ATmega1284P), que dicho circuito se encuentre energizado adecuadamente y se encuentre conectado con el phantom, para mayor información consulte el capitulo 2.

4.1. Verificación del puerto COM asignado.

Al instalar los drivers del phantom, cuando usted lo conecte a la computadora, esta lo detectará como un puerto serie virtual, usted necesita saber que numero de ”COM port” le fue asignado al phantom para darlo de alta en la IDE de ATMEL Studio 6, esto lo puede revisar en el panel de administrador de dispositivos (de click en ***INICIO***, click derecho en ***Equipo*** y seleccione ***Propiedades***, después de click en ***Administrador de dispositivos***).

En la ventana que se abre, y con el phantom ya conectado a la computadora, ubique ***Puertos COM y LPT*** de click para que se extienda los elementos dentro de esa categoría y ubique el que aparece con el nombre de ***USB Serial Port (COMx)***, ese es el que corresponde al phantom (la x representa el numero asignado, en este ejemplo COM5).

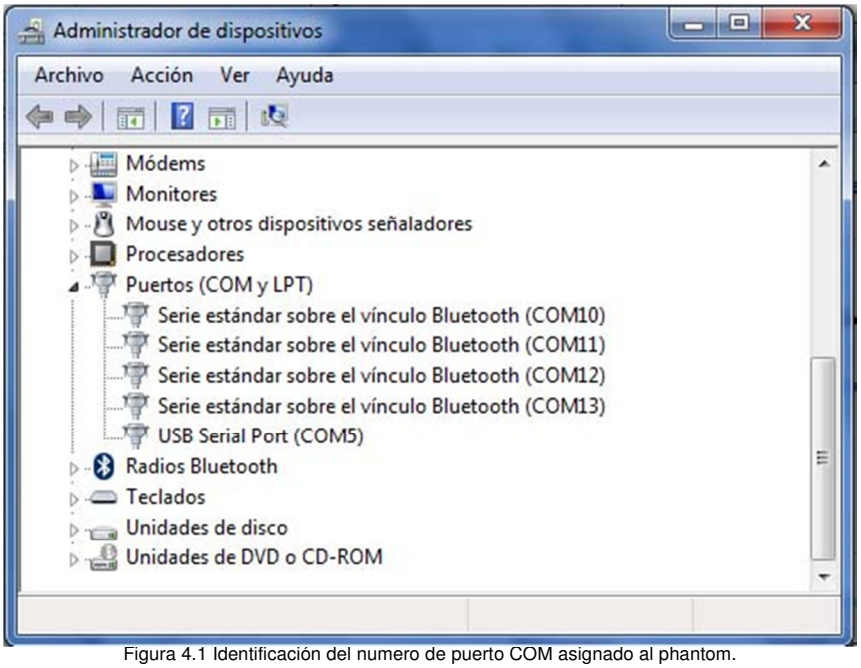


Figura 4.1 Identificación del numero de puerto COM asignado al phantom.

4.2. Conexión con ATMEL Studio 6.

Abra el programa ATMEL Studio 6, con el phantom conectado al equipo, de click en el menú Tools, encontrara la opción Add STK 500, de click en ella y seleccione el puerto COM que corresponde al Phantom.

Figura 4.2 Agregar el phantom al ATMEL Studio 6, como herramienta de programación.

Figura 4.3 Selección del puerto COM asignado previamente al phantom

Una vez hecho lo anterior usted puede acceder a la ventana de programación haciendo click en Device Programming (lo puede ver en la figura 4.2) en el menú Tools o directamente en su icono en la barra de herramientas.

En la ventana que se despliega, seleccione en

- ***Tool***: STK500 (cerciórese de que el COM corresponda con el que esta asignado actualmente a su phantom).
- ***Device***: la matricula del AVR que desea programar
- ***Interface***: ISP[I]

Y de click en ***Apply,*** aparecerán varias opciones en la parte central de la ventana, dependiendo de que requiera hacer deberá seleccionar la opción adecuada, a continuación se da una descripción breve y se ilustra un ejemplo.

Figura 4.4 Selección de herramienta dispositivo e interface de programación.

[I] El phantom solamente soporta la interface ISP, asegúrese de no seleccionar la otra opción (HVPP).

4.3. Lectura de firma de dispositivo.

Este es un paso opcional, pero recomendado, dado que permite verificar que las conexiones eléctricas entre la computadora, el phantom y el AVR a programar, sean adecuadas.

Para leer la firma del dispositivo de click en ***Read***, note que puede aparecer un mensaje (mostrado en la figura 4.5), indicando que el voltaje leído es de 0.0 V, este mensaje es normal, dado que el phantom no lee el voltaje del dispositivo que va a programar.

Device voltage warning
Read voltage 0.0V is outside selected device's operating range: 1.8 V to 5.5 V.
Do you want to continue?
Yes
No
Figura 4.5. Mensaje de advertencia de lectura de 0.0 V

Hecho lo anterior, usted podrá leer la firma en la casilla llamada Device Signature, como se muestra en la imagen 4.6. La firma leída debe ser del estilo 0x1E9XXX, donde XXX representa valores que dependen del AVR que este leyendo. Si algo falla, revise sus conexiones.

Note que la casilla de Voltaje leído cambiará de color a rojo indicando que no pudo leer el voltaje; como se acaba de describir, en las líneas anteriores, esto es normal.

4.4. Programación de la memoria Flash / EEPROM del microcontrolador.

Para descargar el código .hex al microcontrolador, seleccione la opción ***Memories*** ahí se despliegan las opciones tanto de memoria Flash (.hex) como EEPROM (.eep). Recuerde que el programa debe descargarse en la memoria Flash, la memoria EEPROM solo es para datos que se deben retener en memoria no volátil.

En la subsección nombrada Flash, puede dar click en el botón ***"…"*** para buscar el archivo .hex que desee descargar. Una vez hecho esto puede dar click en Program para grabar el código en la memoria del AVR.

Si su proyecto lo requiere y cuenta con un archivo .eep, puede descargar el archivo a la eeprom realizando un proceso similar al señalado para la memoira Flash, esta vez, en la subsección nombrada EEPROM.

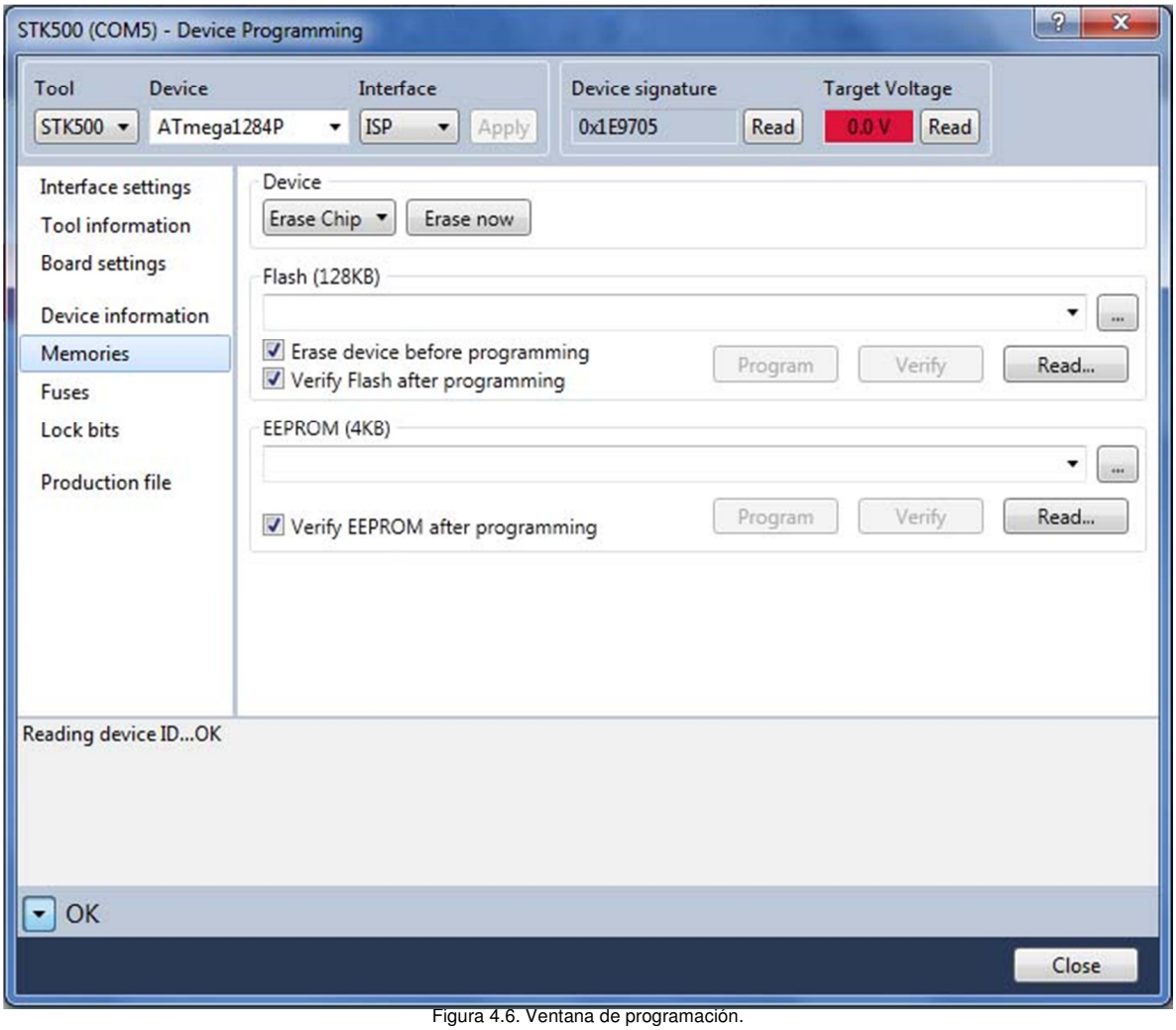


Figura 4.6. Ventana de programación.

4.5. Fusibles y Lock bits.

Para opciones más avanzadas usted puede utilizar las secciones de Fuses o Lock bits. Tenga en cuenta que la modificación de algunos de estos ”bits fusibles” pueden hacer inutilizable el AVR. Antes de realizar cualquier cambio en estos bits refiérase a la hoja de datos correspondiente para obtener la información correspondiente a la funcionalidad de dicho bit.

Los Lock bits proveen capacidades de bloquear ciertas modificaciones a las memorias del AVR, para mayor información consulte la hoja de datos del AVR correspondiente.

4.5.1. Caso de Ejemplo: Modificación de la fuente interna de reloj de un AVR.

Note que cada AVR tiene opciones similares, pero no iguales, en estos ejemplos estamos utilizando un ATmega1284P.

El ATmega1284P cuenta con 2 fuentes de reloj internas y un divisor entre 8, de fábrica se encuentra programado con una fuente de reloj de 8 MHz y el divisor entre 8 activado, por tal motivo la frecuencia a la que opera es de 1 MHz, si queremos cambiarlo para que opere a 8 Mhz debemos hacer lo siguiente:

Vaya la sección Fuses, ubique el ***Fuse name***, llamado ***CKDIV8***, notará que está activada la opción, de click en la casilla para desactivarla. El icono verde al inicio del fusible cambiara a uno amarillo para indicarnos que se ha realizado un cambio, sin embargo esto no implica que se haya aplicado al AVR, para aplicar el cambio debe dar click en el botón de Program de esta misma sección.

La modificación de un fusible no altera el contenido de las memorias Flash ni EEPROM, por tal motivo si ya había descargado una aplicación y desea conservarla no es necesario que reprograma las memorias de AVR.

Como se ha mencionado la modificación inadecuada de algunos de estos bits fusibles puede inhibir la correcta operación del AVR, consulte la hoja de datos del AVR antes de realizar cambios en estos fusibles.

4.5.2. Caso de Ejemplo: Modificación inadecuada de bits fusibles.

Si usted modifica el fusible denominado ***SUT_CKSEL***, usted cambia la fuente de reloj del AVR, si realiza un cambio en ella, digamos, de la configuración que tiene de fábrica: ***INTRCOSC_6CK_65MS*** a otro valor, digamos: ***EXTXOSC_8MHZ_XX_16KCK_65MS*** lo que usted esta indicando es que cambiara a una fuente de reloj externa de 8MHz o superior, que demorará 16K ciclos de reloj + 65 ms para que el AVR comience su operación, si usted no cuenta con dicha fuente externa de reloj conectada al AVR, este quedara inhabilitado para ser usado, leído o reprogramado hasta que le conecte la fuente de reloj que ha especificado (por ejemplo un cristal o un oscilador).

Otro caso también muy importante es el bit ***SPIEN***, por ningún motivo debe modificar este valor, dado que la desactivación de este bit impedirá de manera irreversible (para el phantom) que pueda volver a reprogramar / leer el contenido de las memorias del AVR.